

**IN THE SPECIFICATION**

**Please amend Page 2, Line 21 to Page 3, Line 15, as follows:**

---

AI  
The call control processor executes an application program that performs the basic functions of the base transceiver station, such as call processing, communications protocols, fault management, and the like. Under the control of the application program, the call control processor effectively becomes a plurality of state machines. A state machine is a basic building block of software systems that follow protocols such as call processing, communications protocols, fault management, and other management operations. A state machine as used in this context is described in terms of the following:

State: A place of rest or no change in the state machine.

Event: A stimulus that could cause the state machine to operate and possibly to change to a new state.

Action: An activity performed by the state machine in response to an event.

---

**Please amend Page 21, Line 9 to Page 22, Line 8, as follows:**

---

AI  
cont.  
State machine 310 may retrieve an incoming message from O/S queue 330 in response to the arrival of the incoming message or as part of a scheduled routine. In either event, the task waits on O/S queue 330 until some message is eventually received that triggers the execution of the task (process step 312). Next, state machine 310 translates the message into an event. As mentioned

AB  
Con O.

above, an event is a stimulus that may cause state machine 310 to perform some action and possibly to change to a new state. The translated event is placed in internal queue 324 of state machine 310 for subsequent execution by state machine 310 (process step 314). The execution of the translated event also may cause state machine 310 to generate other new events that are placed in internal queue 324 or that are placed in the internal queue of other state machines of call control processor 310. State machine 310 executes the event(s) in internal queue 324 until internal queue 324 is empty (process step 316). Each event that is executed may cause other events to be generated and placed in internal queue 324. Ultimately, internal queue 324 eventually becomes empty and the task again returns to a state of waiting for a new message to arrive in O/S queue 330.

---

**Please amend Page 24, Lines 6-20, as follows:**

---

AB

There are a number of methods for performing event translation in each state machine and sending the translated event to the correct target queue. These methods include: array, linked list, and hybrid linked list. A linked list or a fixed size array can be allocated to hold the information needed to do the translation and sending of the events. An array has the advantage of being faster than a linked list, but a decision must be made on how many different tasks can request an event as this will determine the amount of memory to allocate. A linked list has the advantage of only taking the amount of memory needed, but performance is lost in order to manage the linked list. An important factor in deciding between a linked list or an array (or some mixture thereof) is the number of events in the state machine that need to be reused by other tasks.

---

**Please amend Page 26, Lines 8-19, as follows:**

ALL  
FIGURE 5 illustrates linked list 500 for performing event translation and directing events to other tasks for use in a finite state machine in a call control processor in accordance with one embodiment of the present invention. Linked list 500 is a list of only the events that need to be shared. These events point to another linked list of other queues and translation events. Some extra overhead is incurred in that every event that comes into a task "walks" the linked list to find out if it is even needed. A first pass optimization may be used to ensure the linked list was sorted so that a determination of whether or not an event was even in the list could be made without going through the entire list each time.

**Please amend Page 26, Line 20 to Page 27, Line 9, as follows:**

175  
Linked list 500 is equivalent to array 400. For example, elements 501-503, elements 531-533 and elements 541-542 are equivalent to array elements A(1,1)-A(1,7) in array 400. If Event 1 (element 501) is processed, state machine 310 is directed by element 502 ("Point To") to elements 531 ("Queue 6") and 532 ("As Event 3"), and subsequently to elements 541 ("Queue 5") and 542 ("As Event 2"). Elements 531, 532, 541 and 542 correspond to array elements A(1,2), A(1,3), A(1,4), and A(1,5), respectively. Advantageously, linked list 500, unlike array 400, does not contain entries for Event 2, Event 3, or Event 6, all of which are undefined.